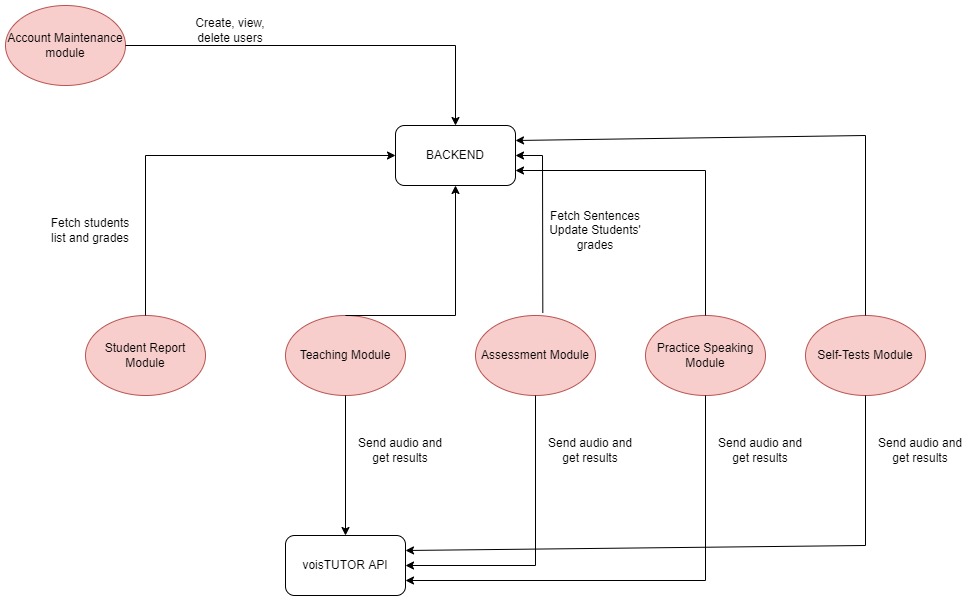
Product Design

|  |  |
| --- | --- |
| **Team** | <Team No: 24, Ayush Agrawal, Janapati Sai Akash, Polavarpu Neeraj, Rahul Mallavarpu > |

# Design Overview

## Architectural design

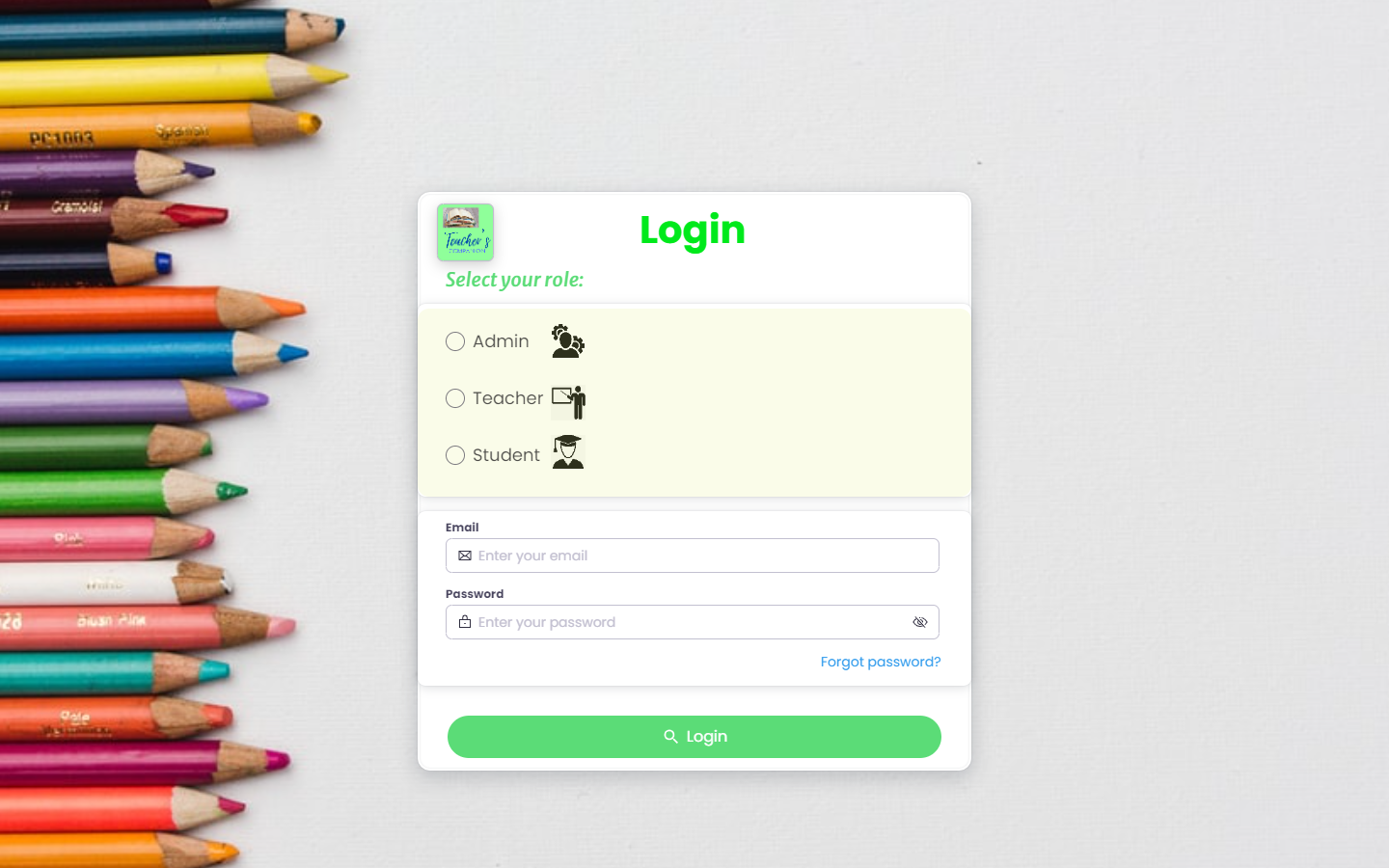
* Student’s Reports Module: To show grade sheet of students and detailed report and analysis of each student which includes all the sentences spoken by him, his average performance chapter wise etc.
* Teaching Module: To teach using selected sentences by recording students’ speech, evaluating with the help of voisTutor API and monitoring class performance
* Assessment Module: To assess the students individually and maintain their grade sheets through in class speech and evaluations
* Practice Speaking Module: To practice speaking sentences by getting feedback from the app along with the option to listen to expert audio
* Self – Tests Module: To evaluate self by taking tests (for students)
* Account Maintenance Module: To add new users, view existing users and remove existing users
* Backend Module: To store all the information in database and provide API endpoints for frontend to fetch data
* voisTutor API : To evaluate speech and provide feedback on various metrics like pronunciation, intonation etc.



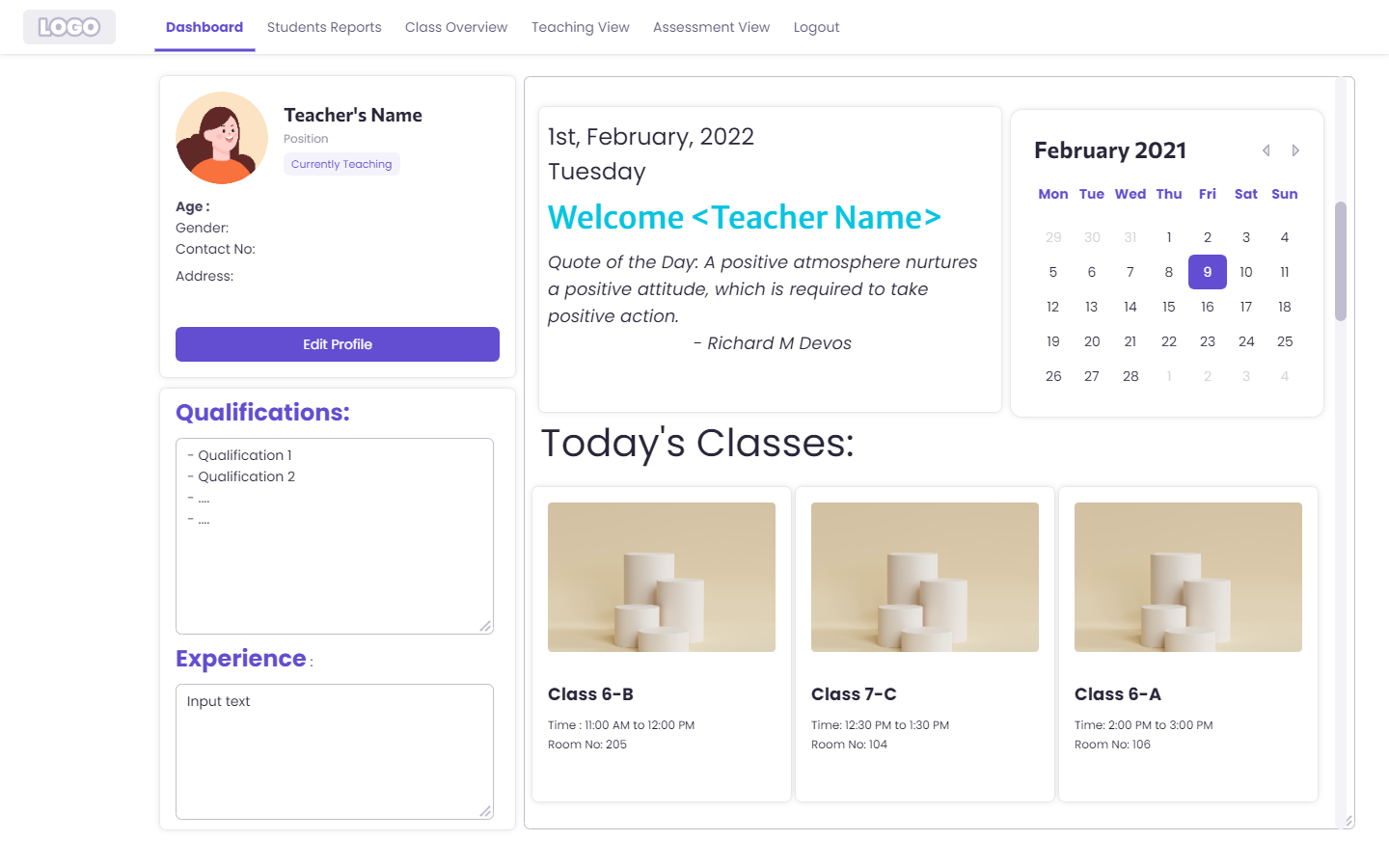
## System interfaces

**User Interface:**

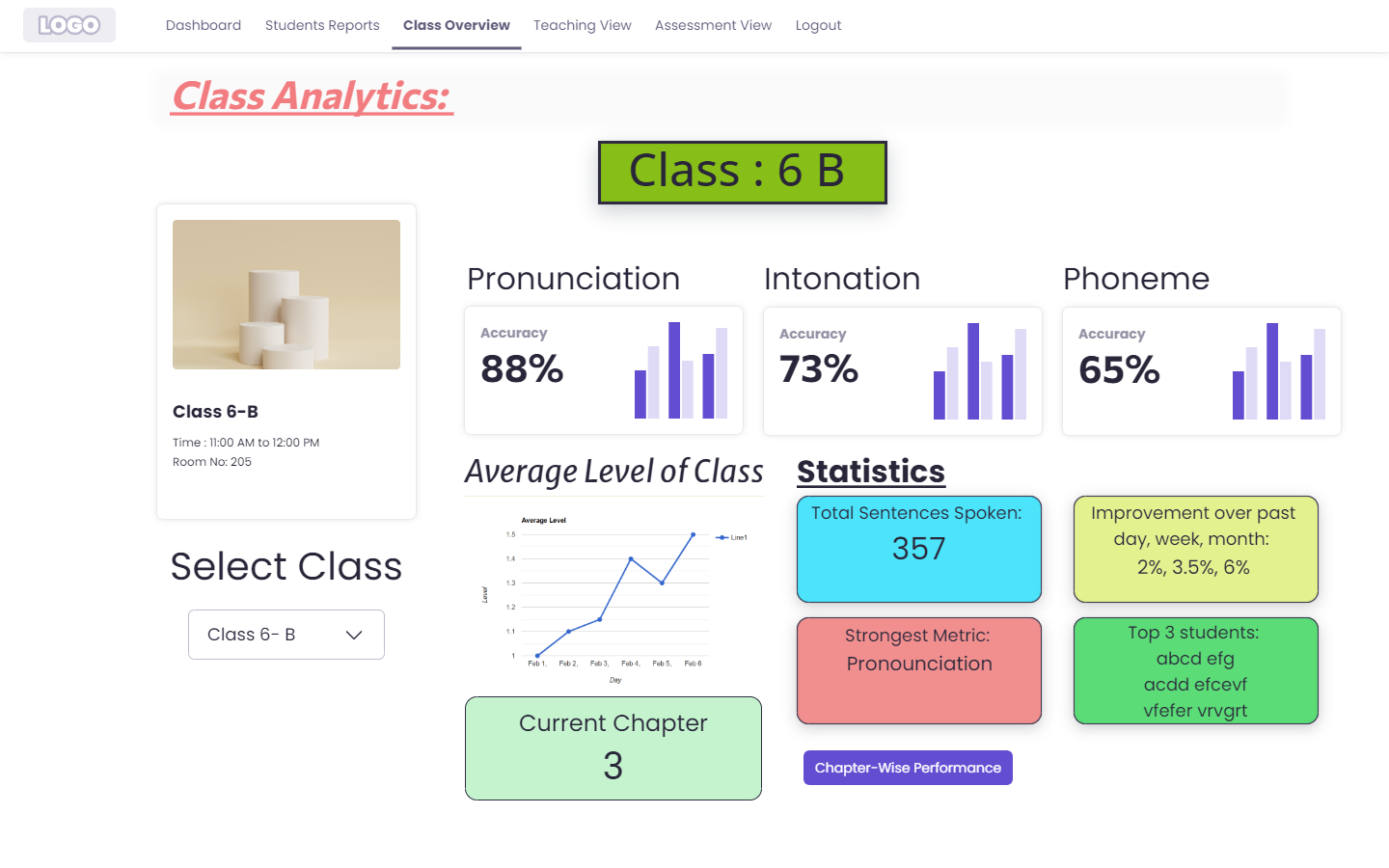
1. First, all users login to the application using a Common login page for all.



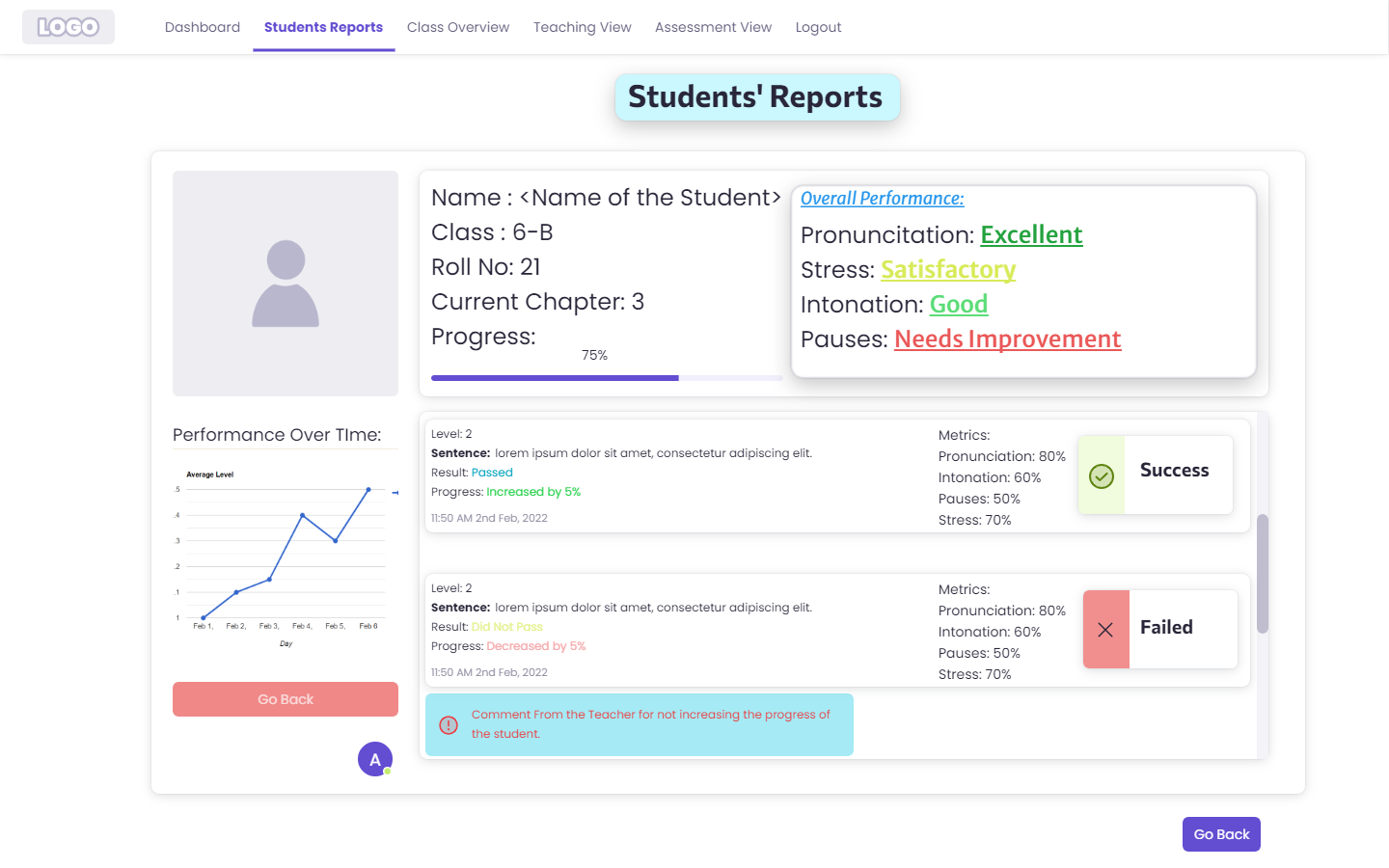
1. Based on his role, he/she will be directed to their respected dashboards



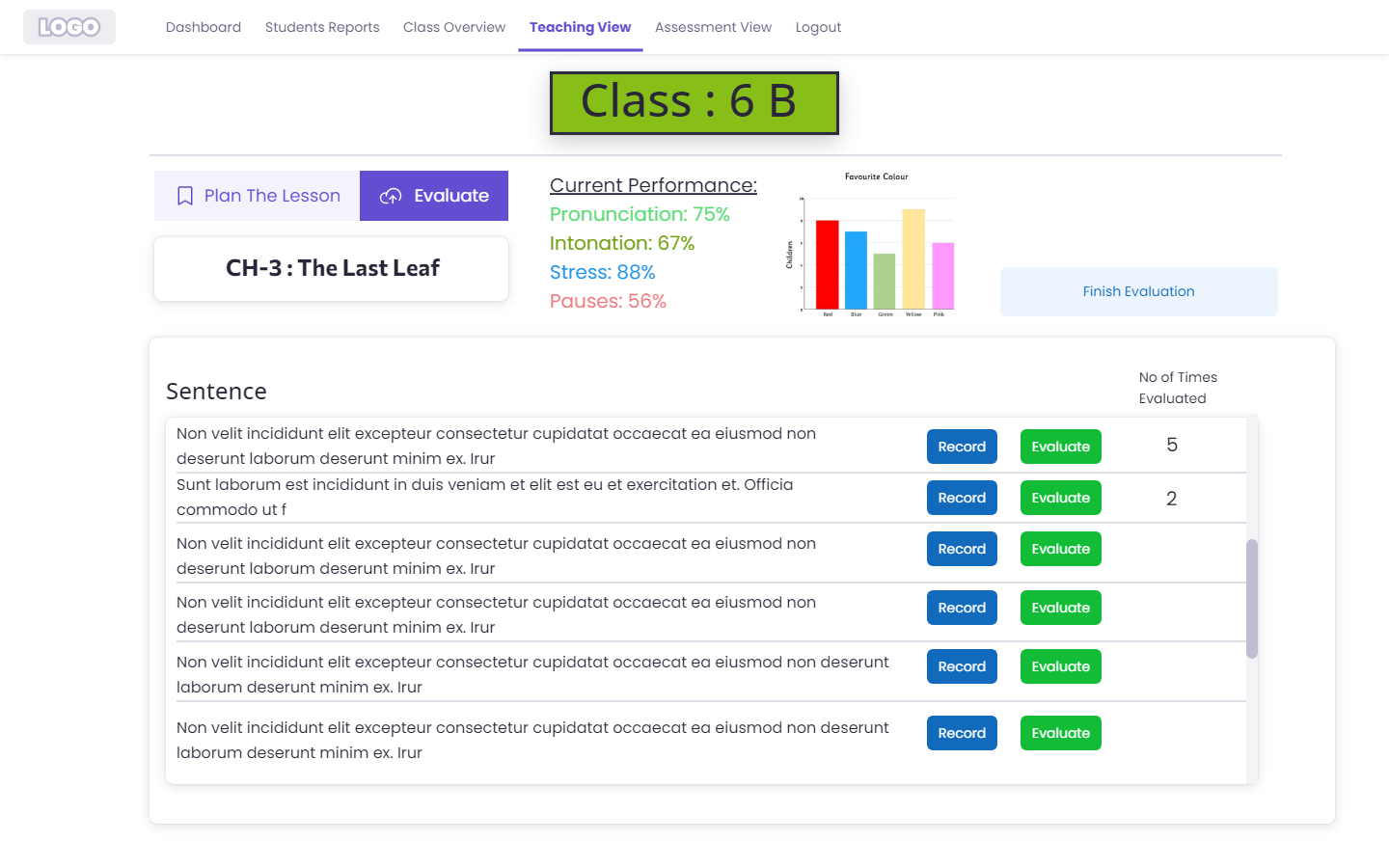
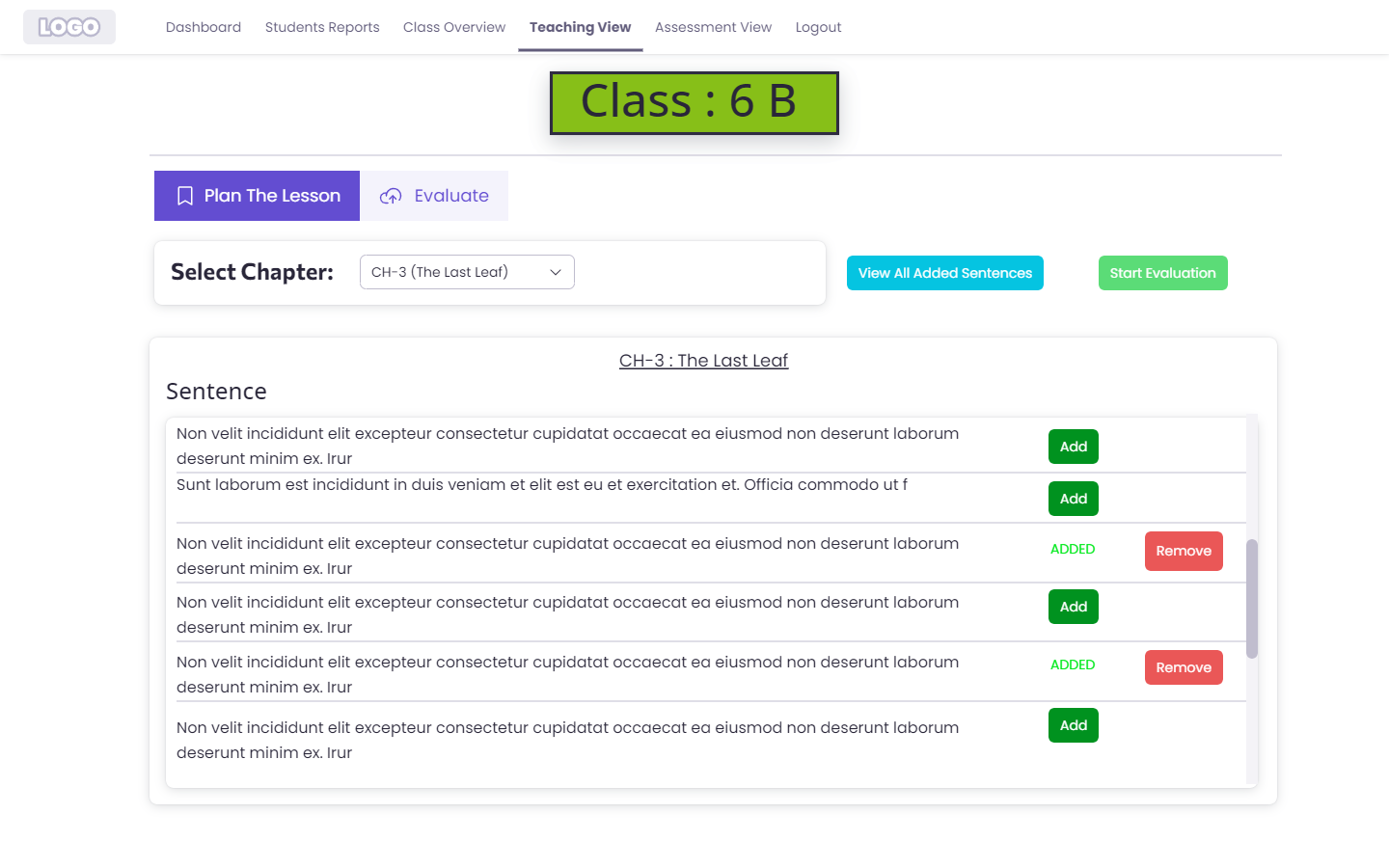
1. For Admin:
2. He will have options to Add new users
3. Similarly, to remove if needed
4. For Teacher:
5. She can view Class performance



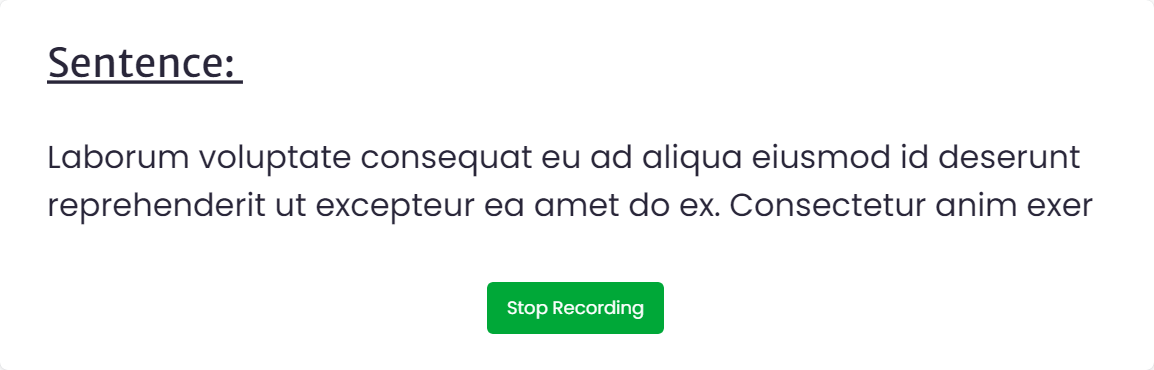
1. She can view Student reports



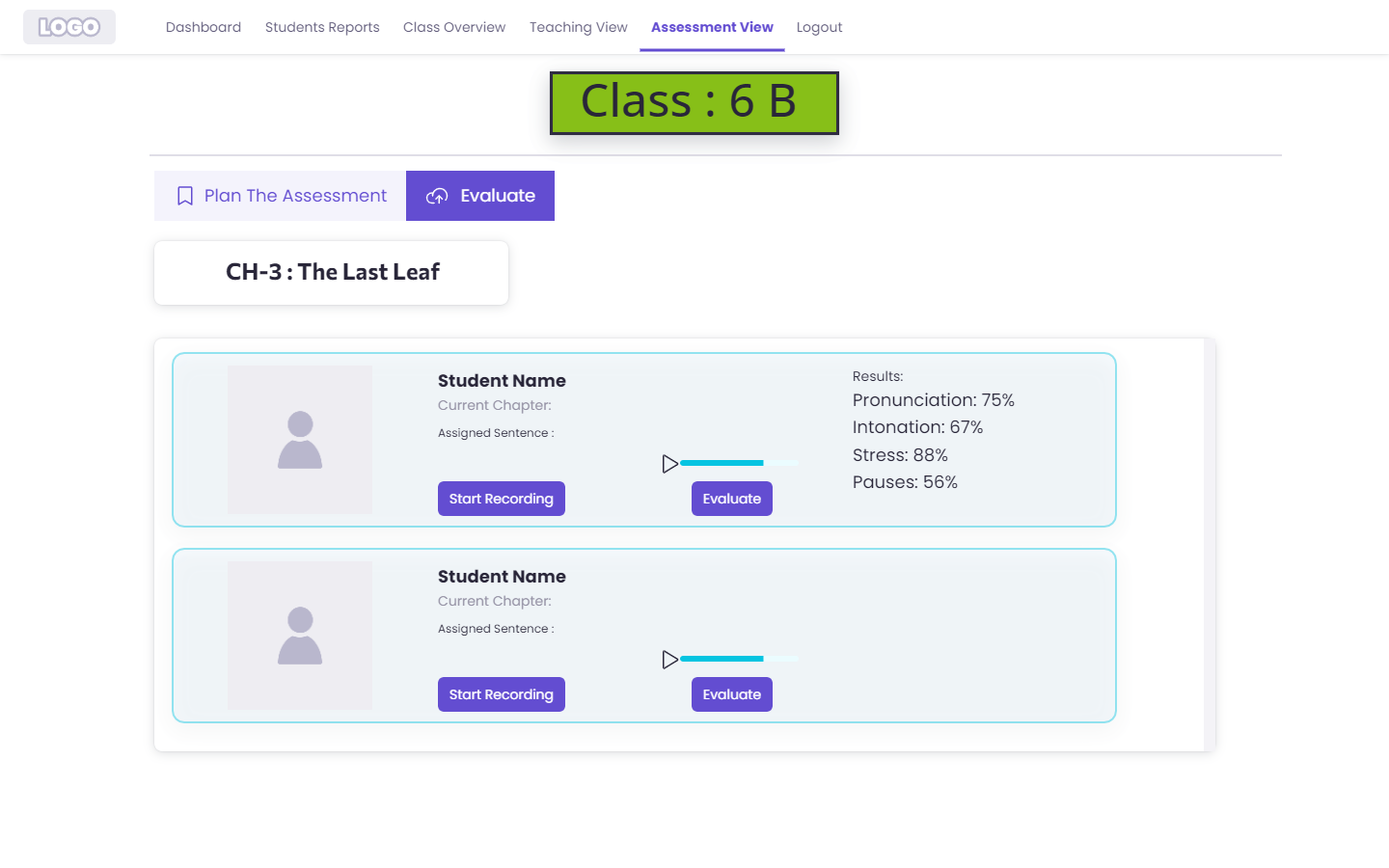
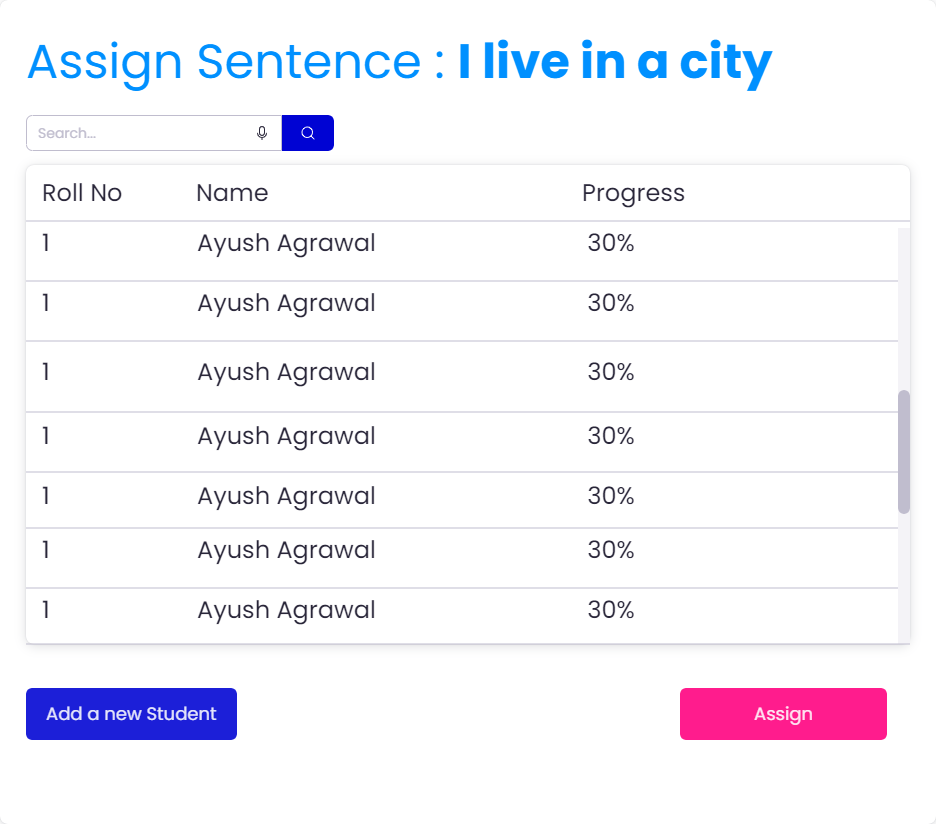
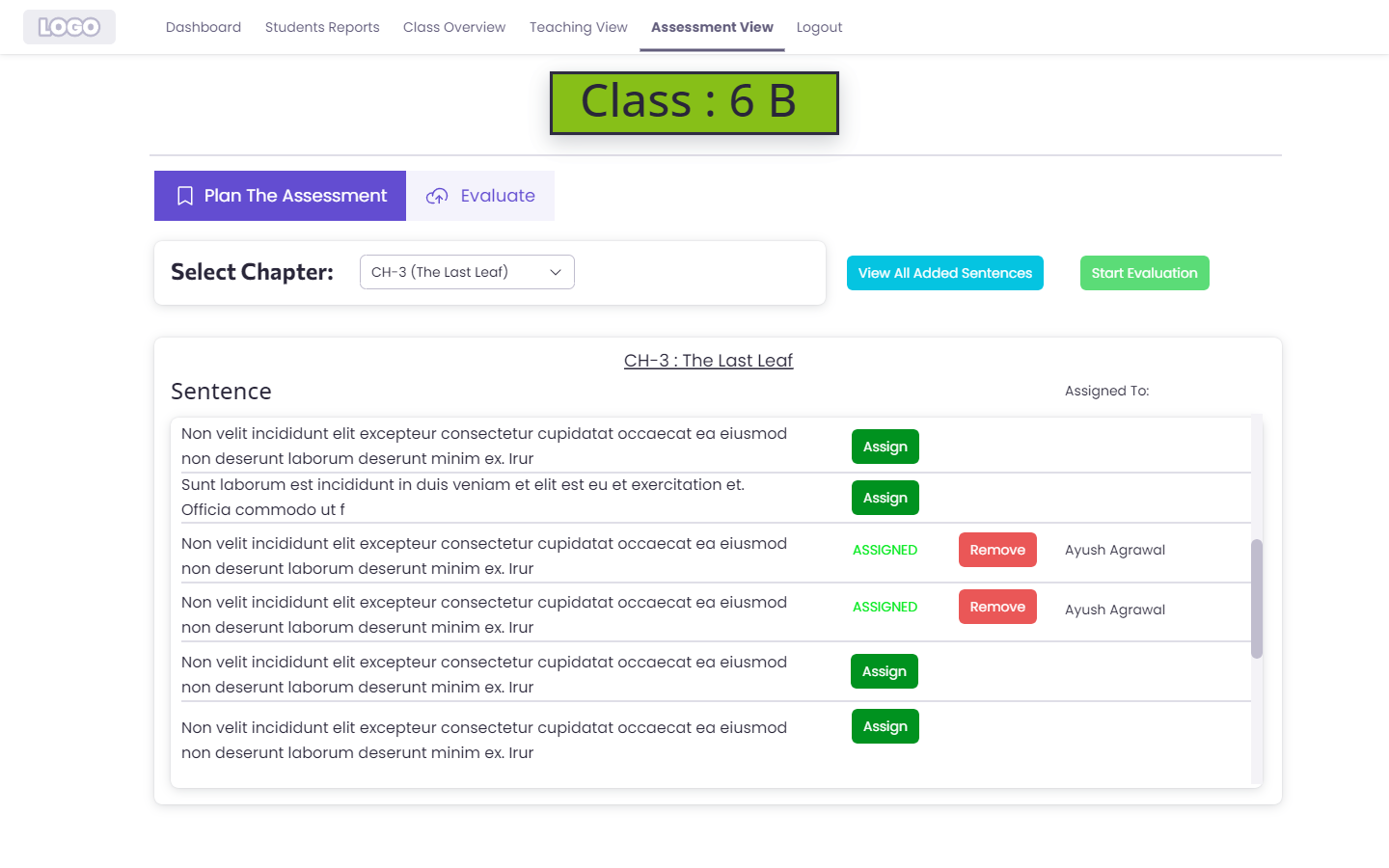
1. She can Plan lesson to teach
2. She can edit details
3. She will have a teaching view, where she have options to evaluate students, modify sentences, and plan the lesson



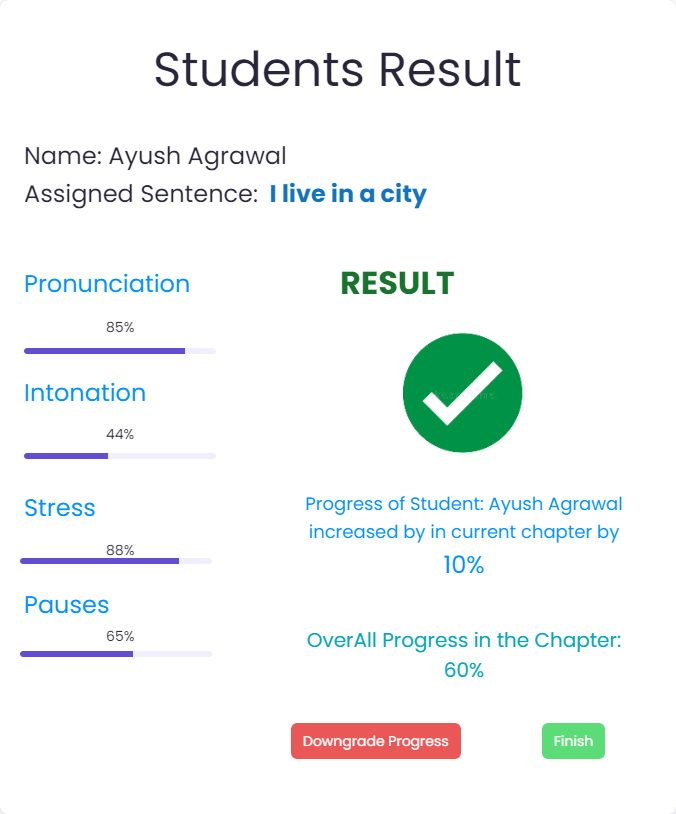
1. In case of evaluation, student will be prompted a sentence on screen to read.



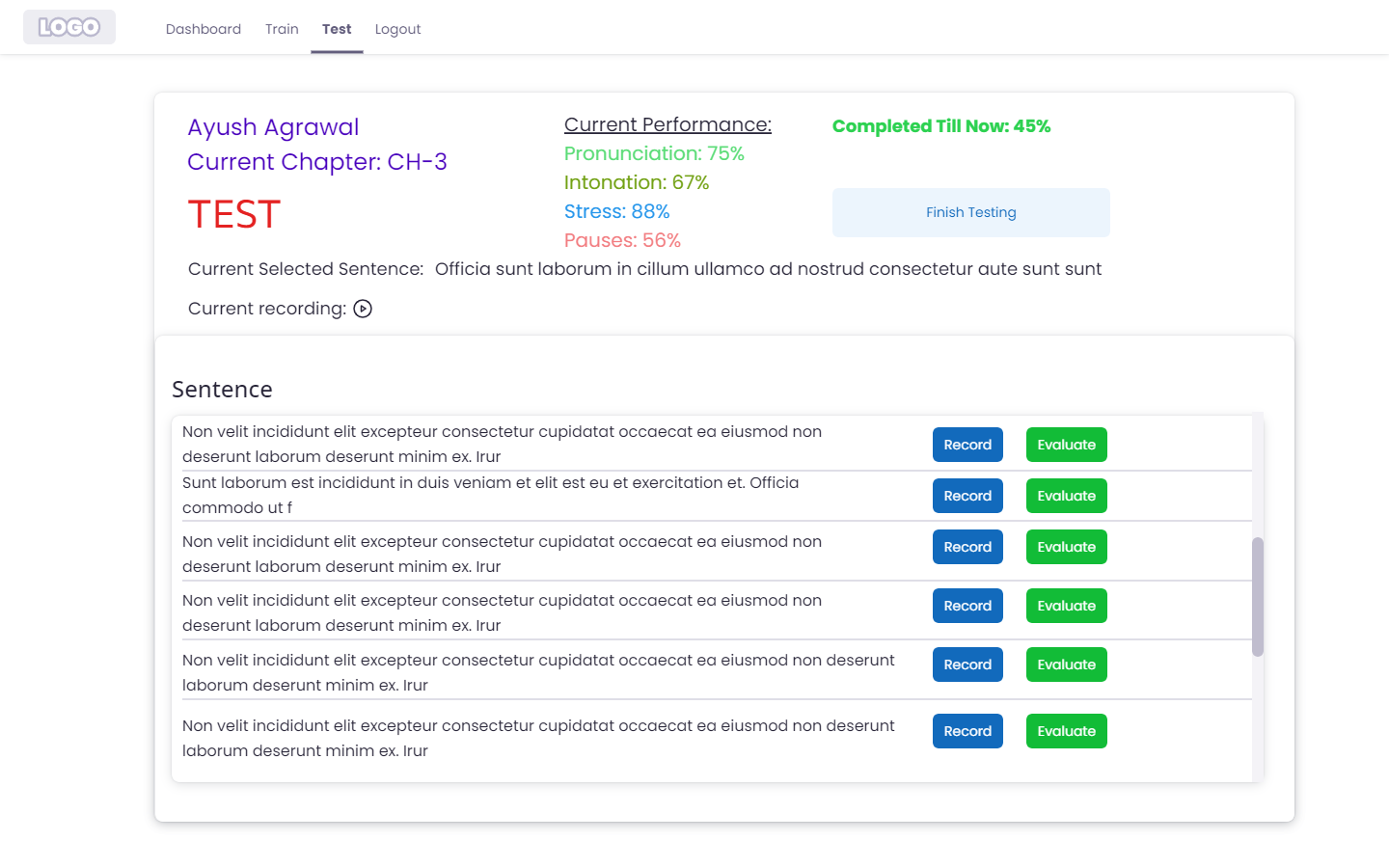
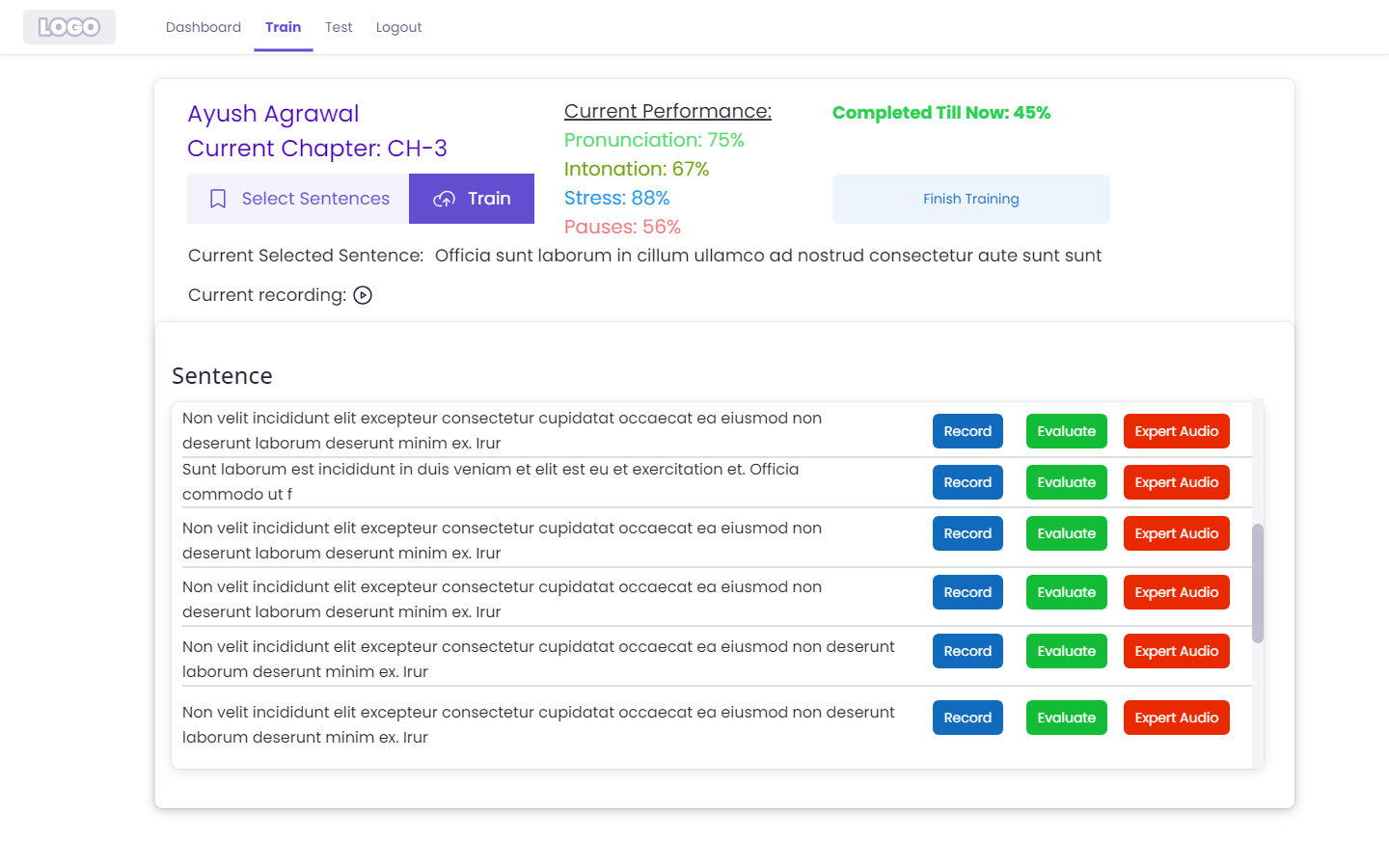
1. On evaluating the recorded audio, results will pop up
2. Assignment view is similar to teaching view with additional features of evaluating specific students.



1. Upon completion of the test, teacher will be displayed the results and prompted with a suggestion to change the level of student



1. For Student:
2. He can train based on his current level of proficiency
3. He can test himself



**APIs**

**Note: app.com is just used to depict the usage of api’s and it is not the final.**

1. **To login: app.com/login.**

**Description: Allows to login for registered users**

**HTTP Method: POST**

**Possible errors:**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **To Register: app.com/register**

**Description: Allows to register new users by Admin**

**HTTP Method: POST**

**Possible errors:**

1. **Teacher:**
2. **View Class: app.com/teacher/:id**
3. **Description: view class report of a particular class**
4. **HTTP Request: GET**
5. **Possible errors**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Class roster : app.com/teacher/roster/:id**
2. **Description: View Class roster of a class**
3. **HTTP Request: GET**
4. **Possible errors**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Edit her details : app.com/teacher/edit/:id**
2. **Description: Edit details of Teacher**
3. **HTTP Request : POST**
4. **Possible errors:**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Teaching view - Evaluate: app.com/teacher/evaluate/:id**
2. **Description: Evaluates recorded audio and displays result**
3. **HTTP Request: POST**
4. **Possible errors:**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Assignment view – Evaluate: app.com/evaluate/:id**
2. **Description: Evaluate the recorded audio of student**
3. **HTTP Request : POST**
4. **Possible errors:**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Assignment view – Update**
2. **Description: Update results based on teacher’s suggestions**
3. **HTTP Request: POST**
4. **Possible errors:**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Student:**
2. **Test: app.com/student/test/evaluate/:id**
3. **Description: Evaluate the recorded audio of student**
4. **HTTP Request: POST**
5. **Possible errors**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Train app.com/student/train/:id**
2. **Description: Evaluate the recorded audio of student**
3. **HTTP Request : POST**
4. **Possible errors**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

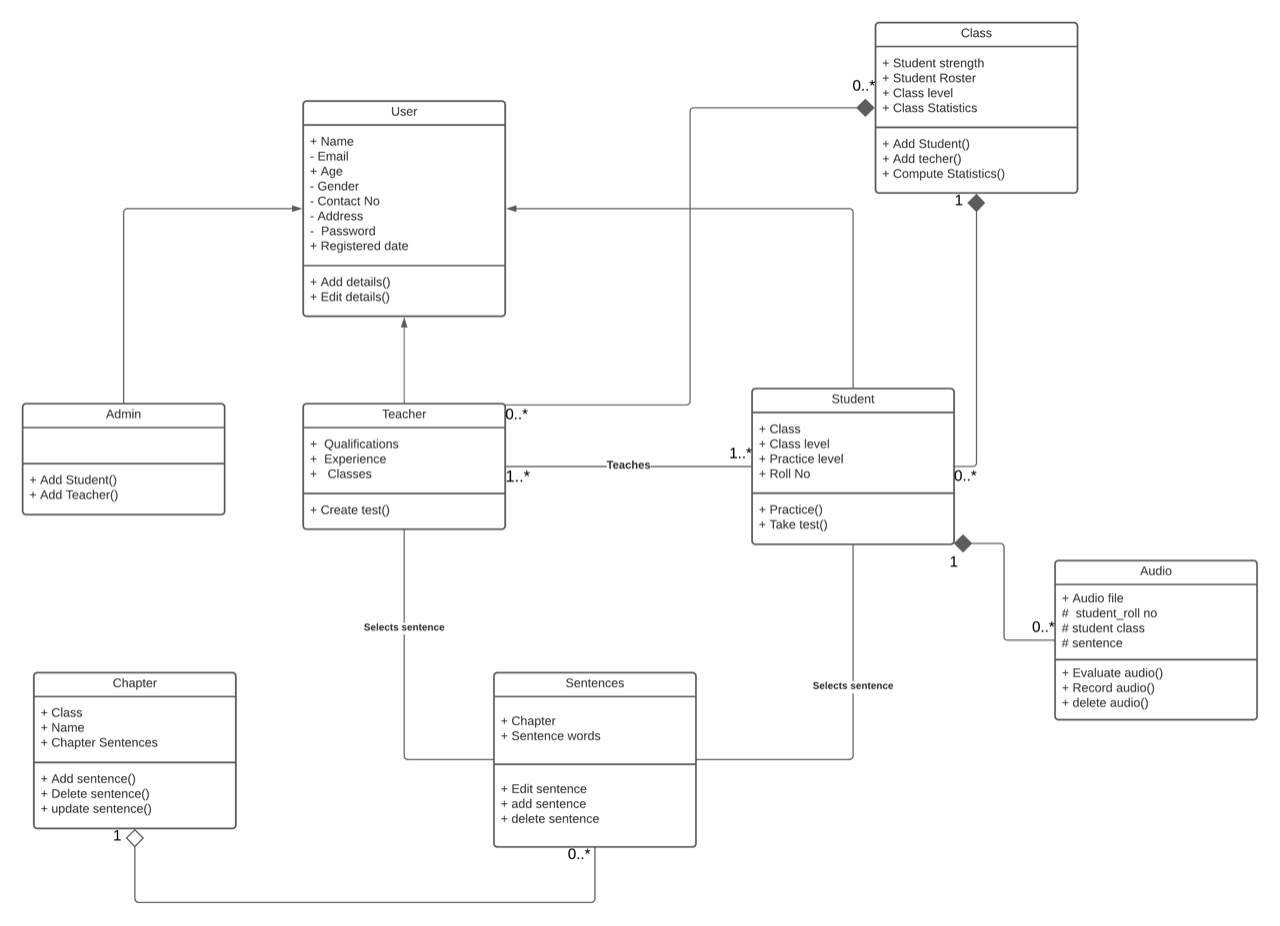
1. **Results : app.com/student/results/**
2. **Description: Show results of Student**
3. **HTTP Request: GET**
4. **Possible errors**

* ERROR 400, If all fields are not present
* ERROR 401 Unauthorized - Access token is invalid or has been revoked.

1. **Logout: app.com/logout**
2. **Description: Logout from application**
3. **HTTP Request: POST**

## Model

.



|  |  |
| --- | --- |
| User | Class state   * Name of the user(admin/teacher/student) * Email of the user(admin/teacher/student) * Age of the user(admin/teacher/student) * Gender of the user(admin/teacher/student) * Contact No. of the user(admin/teacher/student) * Address of the user(admin/teacher/student) * Password of the user(admin/teacher/student) * Reg date. of the user(admin/teacher/student)   Class behavior   * We can add the above details * We can edit the above details |
| Teacher | Class state   * We will have the same information as the user * Also, we will have qualification, experience, class of teacher   Class behavior   * Teacher can create test for evaluating students |
| Student | Class state   * We will have the same information as the user * Also, Class level of the student in that class * Class-section of the student * Practice level of the student which is used for practicing test * Roll\_number of the student   Class behavior   * Can practice from all the chapters * Can take test from the current chapter or specified by teacher |
| Admin | Class state   * We will have the same information as the user   Class behavior   * Admin can add students, teachers new to the school * Admin can edit and delete students, teachers |
| Class | Class state   * We store students strength, student roaster * Class level, which is the level all the students are in. * Statistics of the entire class. (Like: pronounciation, phoneme, intonation)   Class behavior   * Can add student, teacher to the specified class * Compute statistics of the entire class based on pronounciation, phoneme, intonation. |
| Chapter | Class state   * Has class name, chapter name * Stores all the sentences in a chapter   Class behavior   * Can add all the sentences. * Delete sentences * Can edit sentences. |
| Sentences | Class state   * Name of the chapter * All the words of a sentence from a chapter   Class behavior   * Can edit a sentence * Can delete a sentence * Add a sentence |
| Audio | Class state   * Audio file which contains the audio of student recorded * Student details like roll number, class, sentence   Class behavior   * Recording audio() * Deleting audio() * Evaluation audio() |

# Sequence Diagram(s)

